CS4677 Computer Forensics
Web & E-mail Analysis

Chris Eagle

Fall '06

References

- Textbook
  - Chapter 10
    - Browser cache investigation
  - Chapter 11
    - E-mail activity reconstruction
  - Chapter 21
    - Tracing E-mail

Web Activity

- Book discusses IE index.dat files
  - Introduces tools to parse IE data files
  - Useful to learn original URL that file came from
- Mozilla/Firefox
  - Need to find users Cache directory
  - about:cache
  - Use *file* command to identify
- M time – initial download
- A time – last access

Cookies

- Affiliated with particular web site
- Name/Value pair
- Most have expiration time
- IE cookies store creation time
- Browser cookie viewer along with raw dump of cookie file will help you decipher cookie file format

Viewing Email

- Book details various commercial and open source tools
- Easiest solution
  - Use the application that created the data
  - Use different application capable of importing your data
  - Use specialized utilites
    - pst2mbox
    - libPST
  - strings

Tracking E-mail

- Each mail server in chain adds a "Received" header indicating IP address of previous mail server
- Only last (topmost) Received header can be trusted
- Many webmail services add a header containing the originators IP
  - Circumvented by anonymous remailers

Basic Reverse Engineering

Chris Eagle

CS4677, Fall '06

Reading

- Text Chapters 13-15
  - "Files of Unknown Origin" - FOUO

Purpose

- What are the true functionalities and capabilities of a program
- Is there a backdoor or data exfiltration capability built in
- Verify that compiled code matches the source

Static Analysis

- Analyze a program without running it
- Ensures you are not running malicious code
- Most tedious to perform

File Identification

- Use the *file* utility as a first step
  - Indicates platform
    - Windows, Linux, FreeBSD, …
  - Indicates file format
    - ELF, PE, ...
  - Indicates linkage
    - Static, dynamic
  - Whether binary has been stripped

```
proj3_upx.exe: PE executable for MS Windows (console) Intel 80386 32-bit,
               UPX compressed.2.5, dynamically linked (uses shared libs),
               for GNU/Linux 2
proj3a:        ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
               for GNU/Linux 2.2.5, dynamically linked (uses shared libs),
               for GNU/Linux 2.2.5, not stripped,
               corrupted section header size
proj3b:        ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
               for GNU/Linux 2.2.5, dynamically linked (uses shared libs),
               for GNU/Linux 2.2.5, stripped
proj3c.exe:    PE executable for MS Windows (console) Intel 80386 32-bit
proj3d.exe:    PE executable for MS Windows (console) Intel 80386 32-bit
proj3e:        ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
               for GNU/Linux 2.2.5, statically linked, for GNU/Linux 2.2.5,
               not stripped
proj3f:        ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
               for GNU/Linux 2.2.5, statically linked, for GNU/Linux 2.2.5,
               stripped
vxd.vxd:       LE executable for MS Windows (VxD)
echod:         ELF 32-bit LSB executable, Intel 80386, version 1 (FreeBSD),
               for FreeBSD 5.4, dynamically linked (uses shared libs),
               for FreeBSD 5.4, not stripped
fingerd:       ELF 32-bit LSB executable, Intel 80386, version 1 (FreeBSD),
               for FreeBSD 5.4, statically linked, for FreeBSD 5.4, stripped
```

String Content

- Use strings to extract string content
- Make sure you use *strings -a*
- Problems
  - Presence of a string does not imply use of a string
  - Encrypted binaries will show few if any strings

Hex Viewer

- What is this going to tell you?
  - Frankly not much

Object File Parsers

- Some utilities understand the formats of compiled files
  - objdump, nm, ldd, readelf
  - dumpbin
- Display symbols, dependencies and even assembly language listings

Object File Parsing (cont)

- Problems
  - Just because a library is listed does not mean any functions in the library are used
  - Absence of a library does not mean the library is not used
  - The same applies to library functions

Disassembly

- Translate machine language into assembly language
  - Translation to a higher level language like C is called decompilation
    - Far more difficult problem
    - Not a one to one translation problem
- Most sophisticated tool is called Ida Pro
  - Understands many executable file formats and machine languages

Dynamic Analysis

- Involves running the program
- ALWAYS run unknown code in a sandbox environment
  - VMware or machine dedicated to program analysis, disconnected from production network

Program Instrumentation

- When you run a program you will want to see what it does
  - File system changes
    - Filemon – sysinternals.com
  - Registry changes
    - Regmon, Regshot
  - Network traffic
    - Ethereal
  - Library calls made
    - strace, ltrace

Program Control

- Debuggers
  - Allow controlled execution
  - Windows
    - OllyDbg, WinDbg
  - Unix
    - gdb

Compressed/Obfuscated Binaries

- Common among virus/worms/trojans
- Need to analyze unpacked version
- Must uncompress themselves at runtime
  - Use memory dump utility to capture
    - LordPE
  - Use debugger to step through compression
  - Use uncompress feature of compression tool
    - UPX

Code Coverage

- When performing dynamic analysis, how do you know that you have exercised all of the functionality of a binary?
- The percentage of code that you have executed is called the code coverage
- Must craft input in such a way to take all of the conditional branches in the program
  - Extremely difficult to do. Especially without source code